Experiences with 40G End-hosts

Wenji Wu, Liang Zhang, Phil DeMar
FNAL Network Research Group
wenji@fnal.gov, liangz@fnal.gov, demar@fnal.gov

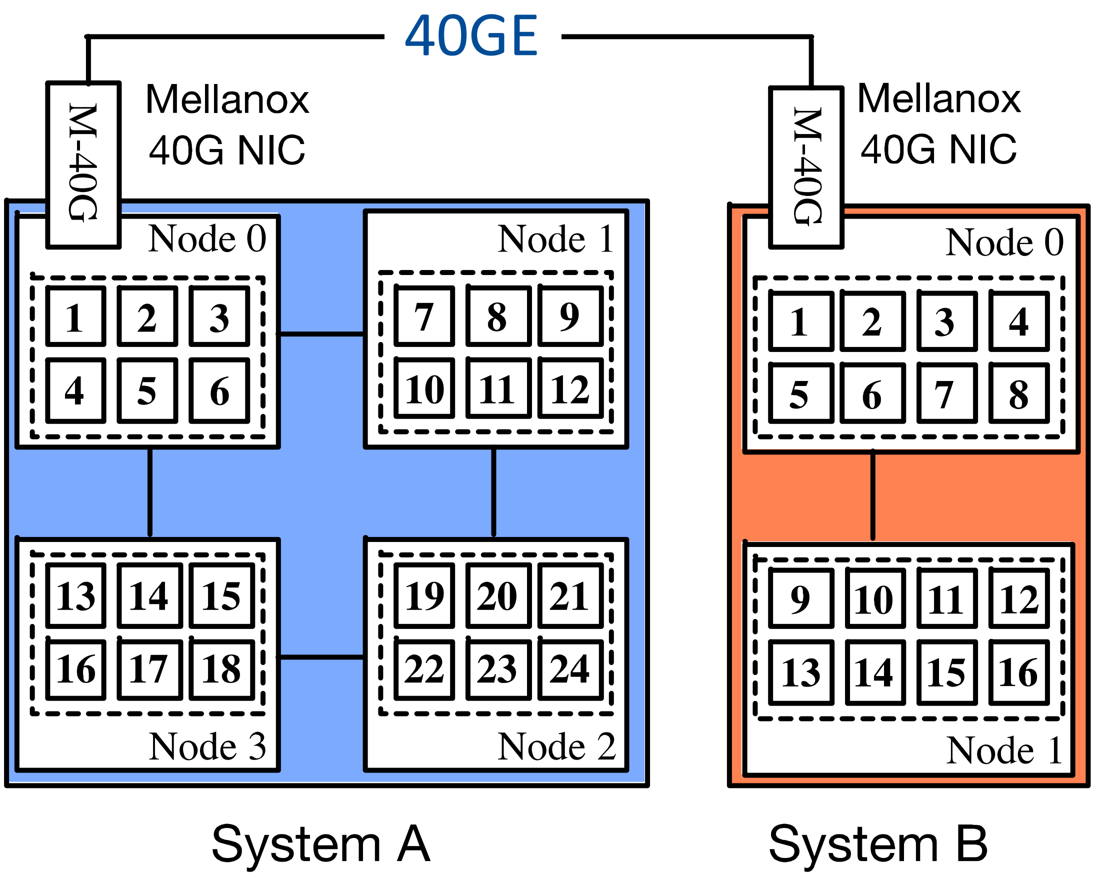
2014 Technology Exchange
October 26 - 31, 2014 Indianapolis, IN

Outline

- Test environment and methodology
  - FNAL 40G System Test Configurations
  - Methodology
- Case 1: Packet drop
- Case 2: I/O locality

Fermilab

FNAL 40G Test Configurations - Hardware



System A

- 4 NUMA nodes
- 24 Intel E5-4607 cores
- 64GB memory
- PCIE-Gen3
- ConnectX®-3 EN 40G NIC

System B

- 2 NUMA nodes
- 16 Intel E5-2680 cores
- 32GB memory
- PCIE-Gen3
- ConnectX®-3 EN 40G NIC

Two systems are connected back to back.

FNAL 40G Test Configurations - Software

- System A:
  - Linux kernel 3.12.23
  - Network stack parameters are tuned
  - Iperf 2.0.5
  - Mellanox driver mlnx-en-2.1-1.0.0
- System B:
  - Linux kernel 3.12.12
  - Network stack parameters are tuned
  - Iperf 2.0.5
  - Mellanox driver mlnx-en-2.1-1.0.0

Methodology

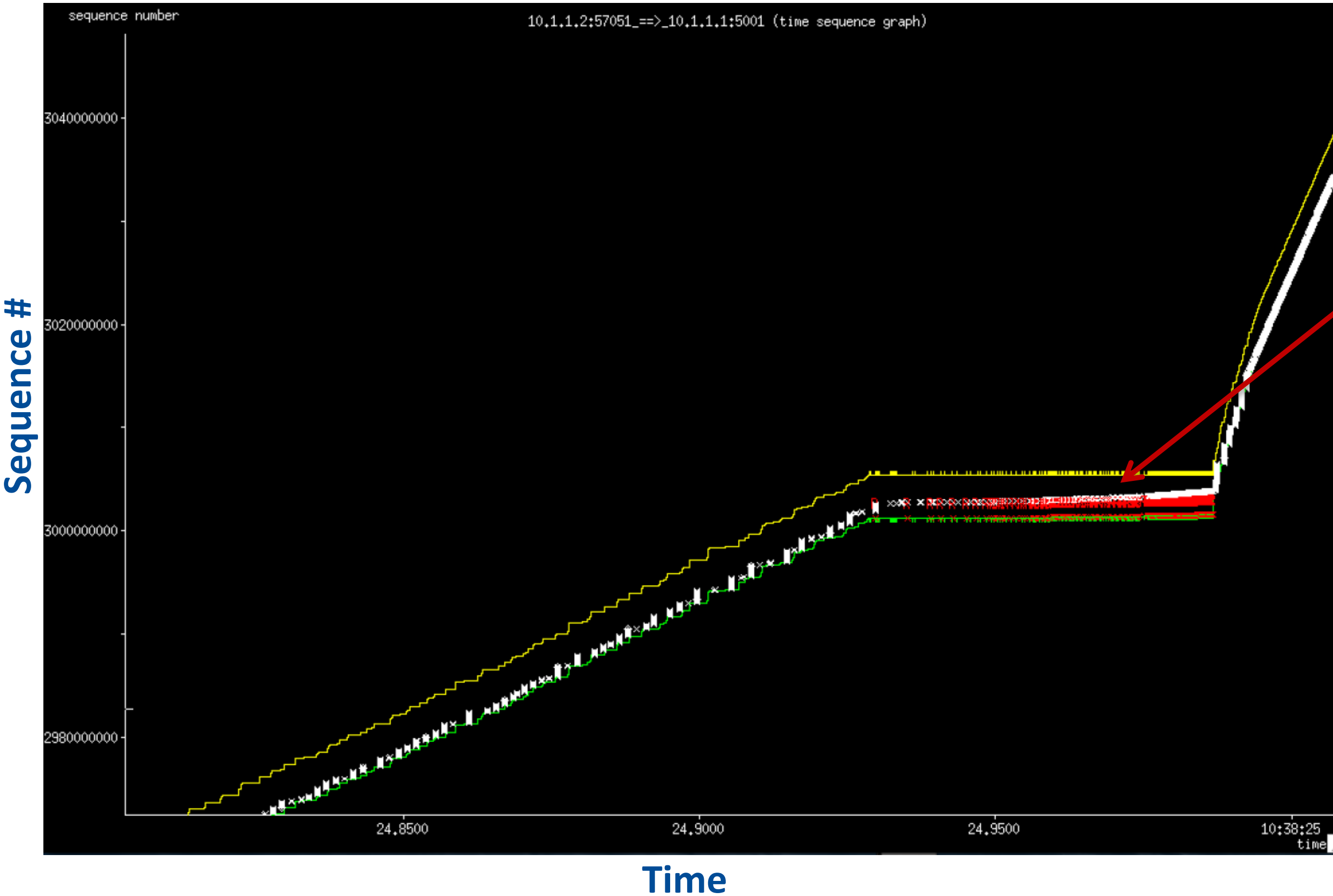
- Run data transfers between System A and B using ***iperf***
- Use ***taskset*** to pin ***iperf*** to specific core(s)
- Use Mellanox adapter IRQ affinity tuning tools
  - http://www.mellanox.com/related-docs/prod_software/mlnx_irq_affinity.tgz
- Use ***tcpdump*** and ***tcptrace*** to capture/analyze packet traces

Case 1 – Packet drop

- Experiment A:
  - Turn off the *IRQ balancer* on both System A and B
  - No *IRQ affinity* tuning on System A and B (Default)
  - Run data transfers with 20 parallel streams from System A to B
  - Run *tcpdump* at System A to capture packet traces

- Experiment B:
  - Turn off the *IRQ balancer* on both System A and B
  - Use *Mellanox IRQ affinity tuning tools* to spread NIC irqs to different cores
  - Run data transfer with 20 parallel streams from System A to B
  - Run *tcpdump* at System A to capture packet traces.

Case 1 – Packet drop (cont.)

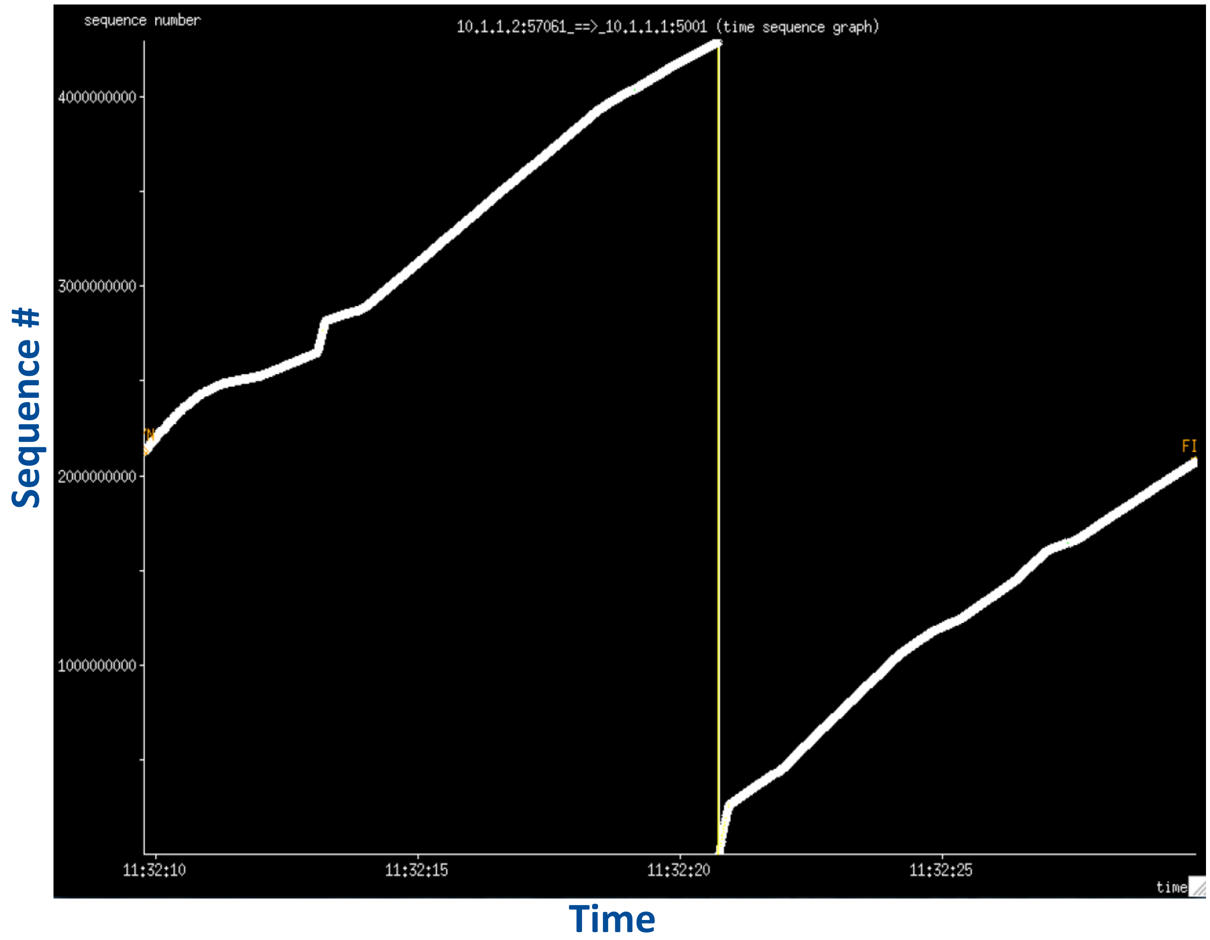


R in read represent packet drops

Significant packet drops!!!

Packet trace of a single stream (Experiment A)

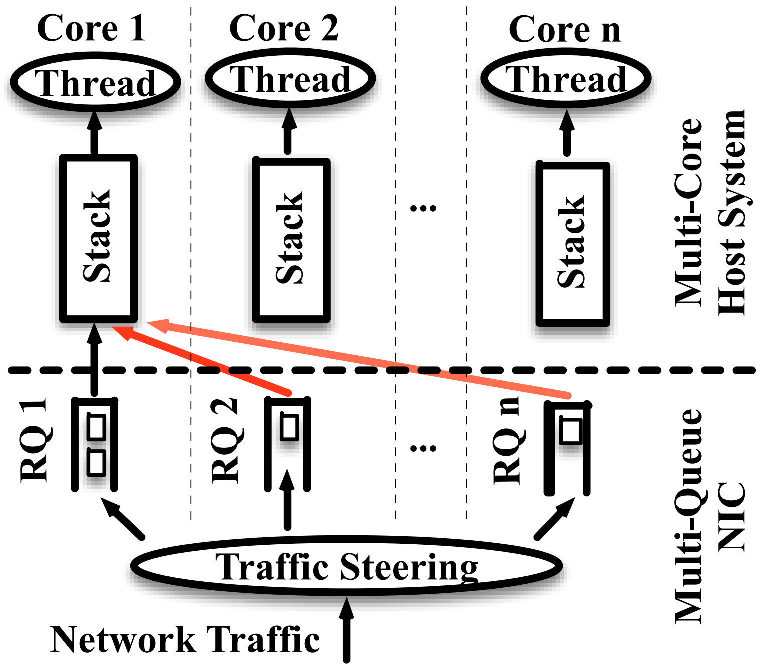
Case 1 – Packet drop (cont.)

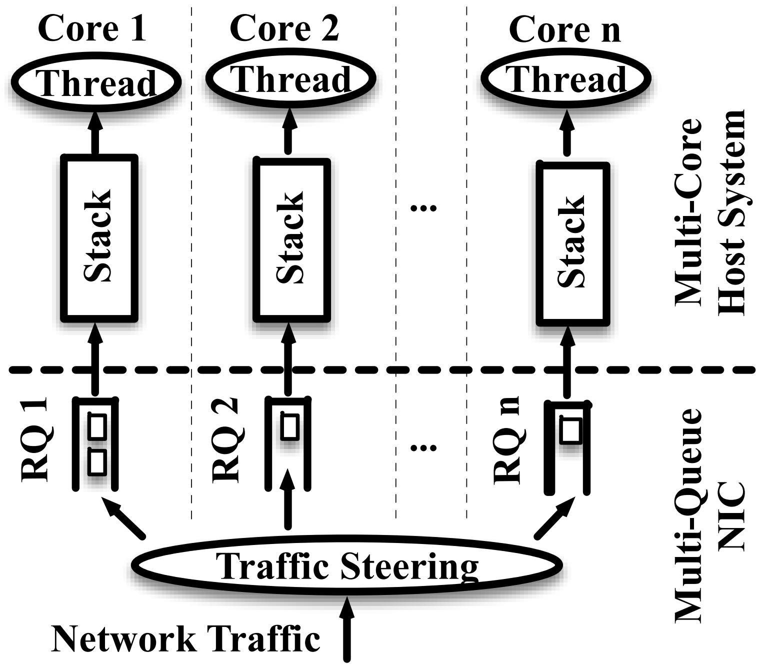


No packet drops are detected !

Packet trace of a single stream (Experiment B)

Case 1 – Packet drop **Why?**





Without Affinity Tuning **With Affinity Tuning**

- Networks are getting faster and CPU cores are not.
- A single core cannot keep up with the high-speed link rates
- We must spread traffic to multiple cores

Case 2 – I/O locality

- Experiment C:
  - Turn off the *IRQ balancer* on both System A and B
  - System A
    - run *Mellanox IRQ affinity tuning tools* to spread NIC irqs to cores on NUMA node 0
    - run "*numactl –N n iperf –s –w 2M*" to pin iperf to NUMA node *n*
      - *n* is varied, ranging from 0-3
  - Run data transfers with single streams from System B to A multiple times

Case 2 – I/O locality (cont.)

```
[root@mdtm-server Downloads]# numactl --hardware
available: 4 nodes (0-3)
node 0 cpus: 0 1 2 3 4 5
node 0 size: 16051 MB
node 0 free: 14592 MB
node 1 cpus: 6 7 8 9 10 11
node 1 size: 16159 MB
node 1 free: 15697 MB
node 2 cpus: 12 13 14 15 16 17
node 2 size: 16159 MB
node 2 free: 15577 MB
node 3 cpus: 18 19 20 21 22 23
node 3 size: 16158 MB
node 3 free: 15745 MB
node distances:
node   0   1   2   3
  0:  10  21  30  21
  1:  21  10  21  30
  2:  30  21  10  21
  3:  21  30  21  10
```

System A has four NUMA nodes
Each NUMA nodes has 6 cores

System A NUMA parameters

Fermilab

Case 2 – I/O locality (cont.)

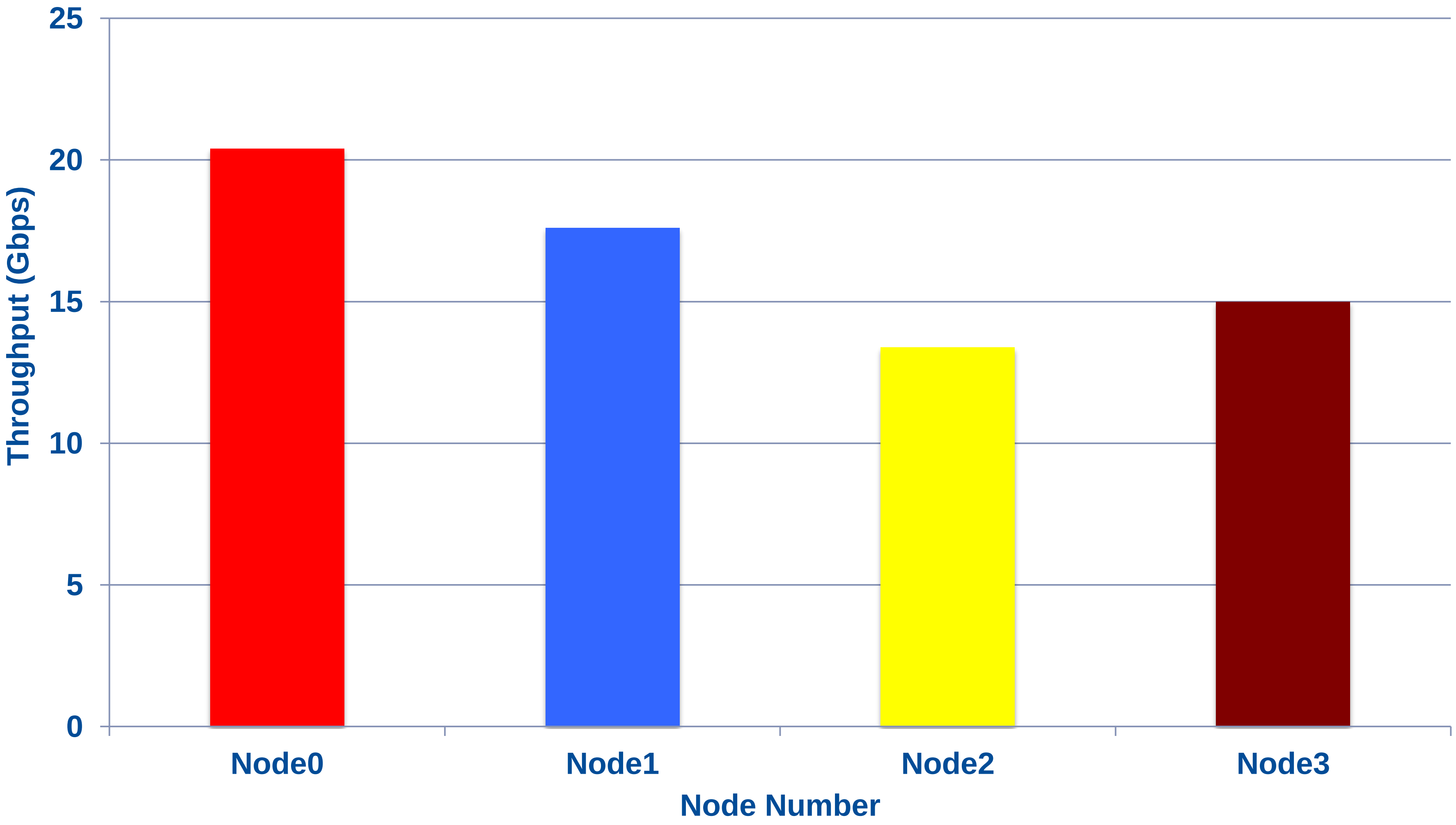
```
[root@mdtm-server Downloads]# ./set_irq_affinity_bynode.sh 0 eth2
-------------------------------------
Optimizing IRQs for Single port traffic
-------------------------------------

Discovered irqs for eth2: 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
Assign irq 272 core_id 0
Assign irq 273 core_id 1
Assign irq 274 core_id 2
Assign irq 275 core_id 3
Assign irq 276 core_id 4
Assign irq 277 core_id 5
Assign irq 278 core_id 0
Assign irq 279 core_id 1
Assign irq 280 core_id 2
Assign irq 281 core_id 3
Assign irq 282 core_id 4
Assign irq 283 core_id 5
Assign irq 284 core_id 0
Assign irq 285 core_id 1
Assign irq 286 core_id 2
Assign irq 287 core_id 3

done.
```

The results of running Mellanox IRQ Affinity tuning tools on System A

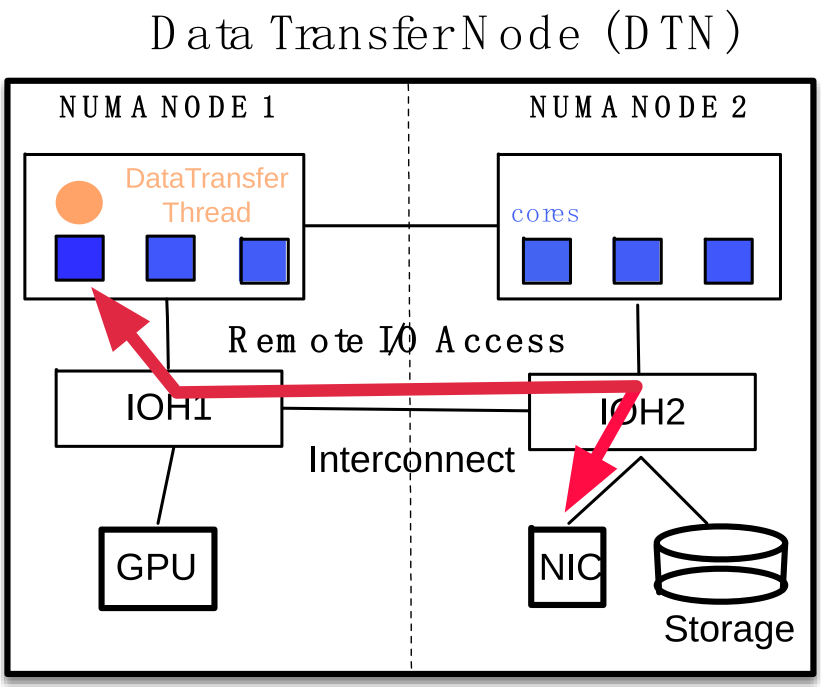
The 40GE NIC is configured with 16 queues
Each queue is tied to a specific core on NUMA node 0

Fermilab

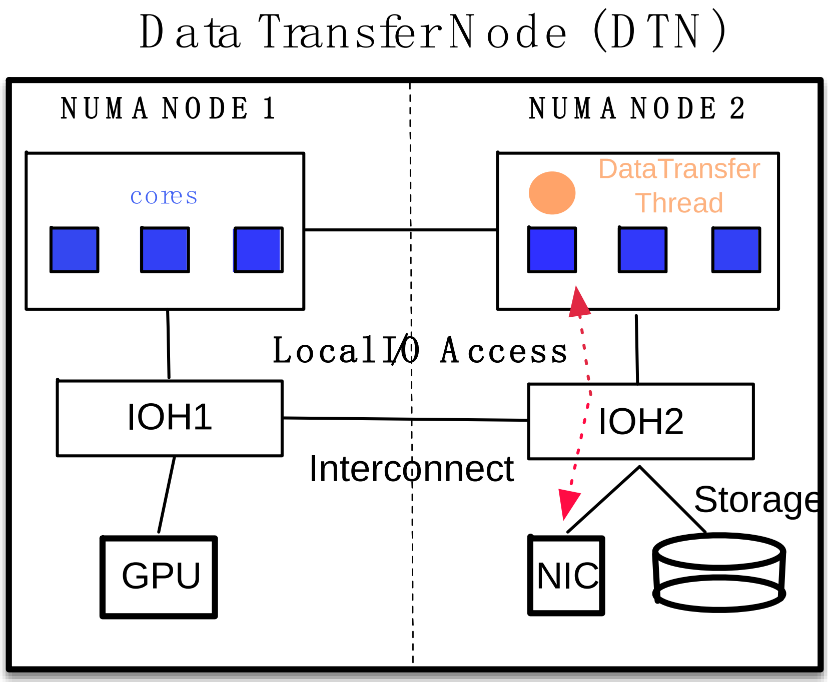
Case 2 – I/O locality (cont.)



Case 2 – I/O locality **Why?**



Data transfer without I/O locality



Data transfer with I/O locality

Remote I/O access is more costly than local I/O access

I/O locality can significantly improves the overall performance